Algorithms

Our design philosophy is **“slow but steady”**: we favor robustness, precise state tracking, and deterministic navigation over speed.

The robot maintains an internal **grid-based position model** and uses **sensor-driven corrections** to remain reliable despite real-world uncertainty.

High-Level Architecture: Dual-Pilot System

The robot operates using a **two-pilot abstraction**, which cleanly separates responsibilities:

- **Pilot 1**: Global navigation and mission planning
- **Pilot 2**: Local sensing, object validation, and pickup

This separation simplifies reasoning and allows tolerance to object displacement or absence.

Pilot 1 — Global Navigation (Grid-Based)

Responsibilities:

- Maintain the robot’s absolute position on the grid
- Navigate to target coordinates (ores, box)
- Handle long-range motion and orientation
- Recalibrate position after each deposit

Key characteristics:

- Uses the robot’s internal grid state
- Accounts for robot orientation
- Stops *just short* of the expected object location
- Assumes an object *should* be nearby, but does not rely on it being exact

Workflow:

1. Initialize robot, motors, sensors, and grid state
2. Select target ore coordinate

3. Navigate to a precomputed stopping point near the ore
4. Hand control to Pilot 2
5. Once Pilot 2 returns, navigate to the box
6. Deposit the object
7. Recalibrate position using box and wall alignment
8. Repeat

From Pilot 1's perspective, **object acquisition is abstracted away** — it simply "goes near ore" and later "has ore" or not.

Pilot 2 — Local Object Acquisition

Pilot 2 takes over once the robot is near the expected ore location.

Responsibilities:

- Precisely locate nearby objects
- Distinguish ores from obstacles
- Perform pickup
- Restore the robot to Pilot 1's expected state

Gyroscope-Based Sweep & Classification

Pilot 2 performs a **controlled angular sweep** using the gyroscope and ultrasonic sensor:

1. Sweep left and right over a defined angle range
2. Record all gyro angles where distance < threshold
3. Compute:
   - Angular span (object width)
   - Average angle (object center)
4. Classify:
   - **Ore**: width within expected bounds
   - **Obstacle / invalid**: otherwise

This makes the system tolerant to:

- Slight ore displacement
- Missing ores

- Non-ore obstacles

Pickup and State Restoration

If the object is classified as an ore:

1. Rotate to the average detected angle (face ore head-on)
2. Advance and grab the object
3. Lift the object
4. Reverse all movements performed during Pilot 2’s control
5. Return control to Pilot 1 at the **exact position and orientation** it expects

If no valid ore is found:

- Pilot 2 backtracks the rotation and passes over a skip order to pilot 1.
- Pilot 1 continues without breaking navigation logic

Recalibration Strategy

After each deposit:

- The robot aligns against the **box and wall**
- Gyroscope reference angle is reset
- Grid position is corrected

This prevents error accumulation over time and ensures long-term positional accuracy.

The navigation system ensures reliability, while Pilot 2 provides adaptability to real-world variance.